**[*Intelligente Parkplatzerkennung mit künstlichen neuronalen Netzwerken*]**

**Projektmanagement-Plan**

[1. Projektorganisation 2](#_Toc4704375)

[1.1. Rollen & Zuständigkeiten 2](#_Toc4704376)

[1.2. Projektstrukturplan 2](#_Toc4704377)

[2. Projektführung 3](#_Toc4704378)

[2.1. Rahmen- und Projektplan 3](#_Toc4704379)

[2.2. Vorbereitung 3](#_Toc4704380)

[2.3. Datenerhebung 4](#_Toc4704381)

[2.4. Programmierung 5](#_Toc4704382)

[2.5. Projektspezifische Aufgaben 6](#_Toc4704383)

[3. Projektunterstützung 7](#_Toc4704384)

[3.1. Tools für Entwicklung 7](#_Toc4704385)

[3.2. Konfigurationsmanagement 7](#_Toc4704386)

[3.3. Tools zur Projektkommunikation und zum Dokumentenaustausch 7](#_Toc4704387)

[4. Qualitäts- und Testplan 8](#_Toc4704388)

[4.1. Definition of Ready und Definition of Done 8](#_Toc4704389)

[4.2. Testdesign & Testautomatisierung 8](#_Toc4704390)

[Anhänge 9](#_Toc4704391)

Versionen:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rev. | Datum | Autor | Bemerkungen | Status |
| 0.1 | 14.03.2019 | Felix Willrich | 1. Entwurf + Eintragen aller Informationen | Abgeschlossen |
| 0.2 | 20.03.2019 | Felix Willrich | Erste Bearbeitung der Rollen | Abgeschlossen |
| 0.3 | 27.03.2019 | Jascha Schmidt | Testfälle beschrieben | Abgeschlossen |
| 0.4 | 28.03.2019 | Frederik Rieß | Arbeitspakete eingetragen | Abgeschlossen |
| 0.5 | 28.03.2019 | Felix Willrich | Gantt Diagramm | Abgeschlossen |

# Projektorganisation

## Rollen & Zuständigkeiten

Zu Beginn des Projektes wurden die Stärken eines jeden Gruppenmitgliedes erörtert. Nachdem die Fähigkeiten, welche innerhalb der Gruppe zu Verfügung stehen, erfasst wurden, wurde ein Projektziel mit einem entsprechenden Produkt definiert. Aufbauend auf die Ziele dieses Projektes, wurde folgende Rollenverteilung innerhalb des Projektes vorgenommen:

|  |  |  |
| --- | --- | --- |
| Name | Rolle(n) | Beschreibung |
| Felix Willrich | Teamleiter, Projektmanagement, Entwickler | Projektleiter vom Projekt, Ansprechpartner vom Kunden,  Entwickler für Teilbereiche |
| Frederik Rieß | Co-Teamleiter, Chefentwickler | Co-Teamleiter vom Projekt,  Koordinierung der Entwicklung |
| Pit-Aurel Ehlers | Entwickler | Entwickler für Teilbereiche |
| Jascha Schmidt | Entwickler | Entwickler für Teilbereiche |

## Projektstrukturplan

Das Projekt wurde in vier Stadien eingeteilt. Die Vorbereitungsphase und die drei Sprints. Der Abschluss des Projekts wird nach dem dritten Sprint durchgeführt. Alle Aufgaben und Arbeitspakete werden in einem GANTT Diagramm dargestellt.

Siehe Anhang 1. Gantt Diagramm

# Projektführung

## Rahmen- und Projektplan

## Vorbereitung

Zunächst wurden innerhalb des Teams wichtige Aspekte bezüglich der Struktur des Projektes und die Anforderungen mit dem Kunden besprochen. Zudem wurde festgelegt, wie untereinander kommuniziert wird, um gegebenenfalls Missverständnisse zu vermeiden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | (Haupt-) Verantwortlicher | Beschreibung | Benötigte Ressourcen | Abhängigkeiten |
| Interview mit Kunden führen | Team | Für die genauen Anforderungen an das Projekt wurde mit dem Kunden ein Interview geführt. | Vereinbarter Zeitpunkt | Kundenwünsche |
| Datenstruktur wählen | Team | Es wurde ein GitHub-Repository und verschiedene Ordner für das Projekt angelegt. | GitHub-Account | Geführtes Kundeninterview |
| Kommunikationsstruktur festlegen | Team | Das Team hat sich intern auf eine Kommunikation über WhatsApp geeinigt. Mit dem Kunden wird über Slack kommuniziert. | Smartphone mit WhatsApp | keine |

## Datenerhebung

In die Datenerhebung fließt das eigentliche Erfassen der Daten mit ein, sowie das Verarbeiten. Es müssen geeignete Quellen für Trainings- und Testdaten gefunden werden. Zudem soll sich mit der Augmentation der Daten beschäftigt und eingeschätzt werden, inwiefern und ob dies umgesetzt werden soll.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | (Haupt-) Verantwortlicher | Beschreibung | Benötigte Ressourcen | Abhängigkeiten |
| Erfassen der Lern- und Testdaten | Jascha Schmidt | Um das CNN lernen zu lassen und das Gelernte zu testen, müssen geeignete Daten benutzt werden. | Internetzugang | Datensatz muss zu finden sein. |
| Bearbeiten und Augmentation der Daten | Felix Willrich | Die Daten müssen obduziert und auf Varianz überprüft bzw. dementsprechend bearbeitet werden. | Vorhandene Daten | Die Daten müssen vorhanden sein. |
| Auslesen der XML-Datei | Pit Ehlers | Zu komplett abgebildeten Parkplätzen sind XML-Dateien mit der Angabe vorhanden, ob die Parkplätze belegt sind. Es muss ein geeignetes Programm entwickelt werden, um diese Daten auszulesen. | Vorhandene Daten | Die Daten müssen vorhanden sein. |
| Label für die Daten | Felix Willrich | Die Daten (Parkplätze) müssen durch geeignete Label gekennzeichnet sein. | Vorhandene Daten | Der Datensatz muss vorhanden sein. |

## Programmierung

In der Teilaufgabe Programmierung werden die Daten zunächst richtig verarbeitet. Anschließend wird ein geeignetes Modell entworfen, um diese Daten für das Training des neuronalen Netzes zu benutzen. Dabei handelt es sich zunächst um ein initiales Modell, bei dem die Hyperparameter in den folgenden Sprints noch optimiert werden sollen. Eine grafische Darstellung soll zudem die Ergebnisse präsentieren.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | (Haupt-) Verantwortlicher | Beschreibung | Benötigte Ressourcen | Abhängigkeiten |
| Einlesen der Daten | Frederik Rieß, Felix Willrich | Es ist eine geeignete Methode zu finden, die Daten einzulesen. | Vorhandene Daten, Jupyter Notebook mit Frameworks | Die Daten müssen vorhanden und vernünftig abgelegt sein. |
| Skalieren der Daten | Frederik Rieß, Jascha Schmidt | Die Daten müssen für Keras entsprechend skaliert bzw. geparsed werden. | Vorhandene Daten, Jupyter Notebook mit Frameworks | Die Daten müssen vorher eingelesen sein. |
| Modell für das neuronale Netz entwerfen | Frederik Rieß, Pit Ehlers | Für das neuronale Netz muss ein geeignetes Modell entworfen werden. Dabei ist die Auswahl der Layer von großer Bedeutung. | Skalierte Daten, Jupyter Notebook mit Frameworks | Die Daten müssen skaliert sein. |
| Lernprozess initiieren | Team | Für die Layer sind initiale Optimizer und eine Loss Function auszuwählen. | Skalierte Daten, Jupyter Notebook mit Frameworks | Ein Modell muss vorhanden sein. |
| Grafische Darstellung der Ergebnisse | Frederik Rieß | Die Ergebnisse sollen grafisch dargestellt werden können. | Fertiges neuronales Netz, Jupyter Notebook mit Frameworks | Das neuronale Netz sollte fertig programmiert sein. |

## Projektspezifische Aufgaben

Unter dieser Teilaufgabe wird die gesamte Dokumentation während der einzelnen Aufgaben und Arbeitspakete verstanden. Zudem sollen nach dem ersten Sprint ein Review gehalten und das neue Sprint Backlog entworfen werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | (Haupt-) Verantwortlicher | Beschreibung | Benötigte Ressourcen | Abhängigkeiten |
| Dokumente erstellen | Team | Strukturen, Verfahren und Ergebnisse müssen dokumentiert werden. | Schreibprogramm auf dem PC | keine |
| Sprint-Review | Team | Review des vergangenen Sprints | Ergebnisse des Sprints | Fertige Dokumente |
| Sprint Backlog | Team | Sprint Backlog des neuen Sprints erstellen | Schreibprogramm auf dem PC | Vorheriges Sprint-Review |

# Projektunterstützung

## Tools für Entwicklung

Während des ersten Projektmeetings wurden folgende Tools für die Entwicklung beschlossen:

|  |  |
| --- | --- |
| Name | Beschreibung |
| Python 3.X | Grundlage jeglicher Programmierung |
| Keras | Paket für maschinelles Lernen in Python geschrieben |
| NumPy | Paket, um das Rechnen mit Matrizen und Vektoren zu vereinfachen, Geschrieben in NumPy und wird passiv mitgenutzt |
| Matplotlib | Library für die Darstellung von Diagrammen |
| Tensorflow | Keras stützt sich auf Tensorflow |
| Jupyter Notebook | Sozusagen die IDE |
| Foto Datenbank | <https://web.inf.ufpr.br/vri/databases/parking-lot-database/>  Grundlage zum Anlernen |

## Konfigurationsmanagement

Im initialen Projektmeeting wurde festgelegt, dass keine Konfigurationen einheitlich eingesetzt werden. Es wird versucht eine möglichst breite Masse an Einstellungen auszutesten um das bestmöglichste Ergebnis zu erreichen.

## Tools zur Projektkommunikation und zum Dokumentenaustausch

Während des ersten Projektmeetings wurden folgende Medien beschlossen, die zur Kommunikation und zum Austausch dienen sollen:

|  |  |  |
| --- | --- | --- |
| Name | Link(ggf.) | Beschreibung |
| Github | https://github.com/Scantraxx123/Parkplatzerkennung | Austausch aller Dokumente und Code |
| Slack |  | Kommunikation mit Kunden |
| Whatsapp |  | Kommunikation untereinander |

# Qualitäts- und Testplan

## Definition of Ready und Definition of Done

Definition of Ready (DoR): Was muss ein Backlog Item mindestens erfüllen, damit es im nächsten Sprint Planning ausgewählt werden kann?

* Alle Teammitglieder müssen die Aufgabe verstanden haben
* Die Aufgabe wird als sinnvoll betrachtet
* Die Aufgabe unterstützt das Sprintziel
* Aufwand der Aufgabe wurde verhältnismäßig kalkuliert
* Ziel ist realistisch

Definition of Done (DoD): Was muss ein Backlog Item erfüllen, damit Sie am Ende eines Sprints vom Product Owner als fertig deklariert werden kann?

* Alle Teammitglieder sehen die Aufgabe als erledigt
* Item wurde getestet
* Ergebnis wird als hinreichend betrachtet
* Das Ergebnis ist in der Versionsverwaltung (Github) hochgeladen
* Ergebnis wurde von anderen Entwicklern überprüft
* Ergebnis ist im Gesamtkonzept mit eingebaut

## Testdesign & Testautomatisierung

Das Projekt umfasst ein mit Hilfe des Frameworks Keras erstelltes neuronales Netzwerk (CNN). Keras bietet zusammen mit der Library Matplotlib die Möglichkeit, diverse Diagramme, wie zum Beispiel die Erkennungsrate, darzustellen. Dies wird die Grundlage der Tests sein, da somit nachvollzogen werden kann, wie genau die Parkplatzerkennung funktioniert, um unter anderem Overfitting zu vermeiden. Weitere Tests werden mit einzelnen Bildern vollzogen, die in den Testdaten liegen. Diese Testdaten zeigen eine Aufschlüsselung, wie viele Parkplätze frei sind. Diese werden mit einem Python-Skript ausgelesen und gegengeprüft durch das CNN.

# Anhänge

1. Gantt Diagramm

